# nginx+nginx-rtmp-module+ffmpeg搭建流媒体服务器服务器

## **前言**

今天给大家分享一篇利用nginx来搭建流媒体服务器，关于Nginx的安装可以看我之前的文章，我之前之所以会写nginx的文章是有一定的道理的，过段时间我会写一些协议栈的学习总结分享：

# 《nginx之静态资源访问和负载均衡的使用！》:

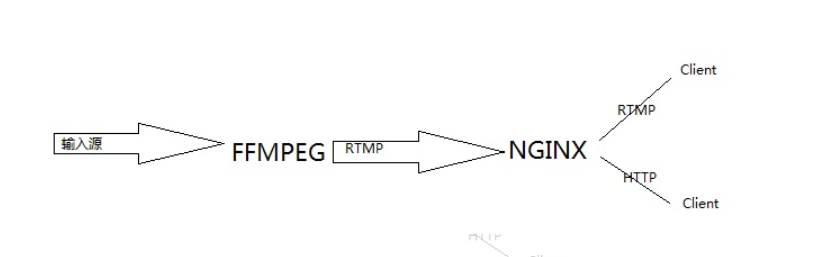
<https://mp.weixin.qq.com/s/71PH3eApz5zrjHXeyuRU5Q>

不同的是，我们这里需要添加一个模块：configure的时候需要增加nginx-rtmp-module的支持,下载好nginx-rtmp-module后解压,然后nginx安装时增加这个模块(--add-module),其它都是一样的。

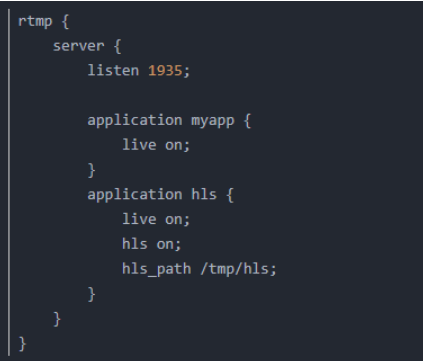
## **一、实际操作：**

### **1、原理：**

nginx配合ffmpeg做流媒体服务器的原理是: nginx通过rtmp模块 提供rtmp服务, ffmpeg推送一个rtmp流 到nginx, 然后客户端通过访问nginx 来收看实时视频流。HLS也是差不多的原理,只是最终客户端是通过HTTP协议来访问的,但是ffmpeg推送流仍然是rtmp的：



### **2、配置conf文件：**

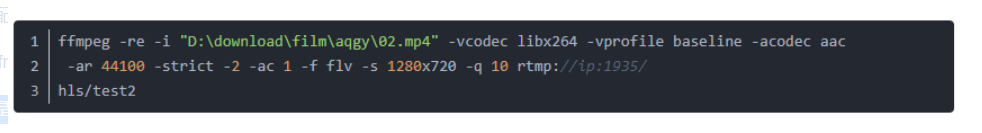


这是一个最简单,最基础的配置。 rtmp监听1935端口,如果是hls的话用hls on开启hls,并且为hls设置一个临时文件目录hls\_path /tmp/hls; 其它更高级的配置可以参看nginx-rtmp-module的readme,里面有比较详细的介绍其它的配置,并且它还提供了一个通过JWPlayer在网页上播放的例子：

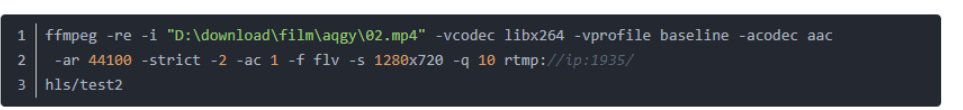
保存完配置文件后,启动nginx,通过netstat -ltn命令可以看到增加了一个1935端口的监听.8080是nginx默认的http监听端口。

然后用ffmpeg推流到nginx:

第一个是推到了上面配置的myapp上:



第二个推流到hls上：



上面这两个流的地址分别是:

第一个就是推送的地址: rtmp://serverIp:1935/myapp/test1

第二个是HTTP地址: http://serverIp:8080/hls/test2.m3u8

* HLS:

HTTP Live Streaming（HLS）是苹果公司(Apple Inc.)实现的基于HTTP的流媒体传输协议，可实现流媒体的直播和点播，相对于常见的流媒体直播协议，例如RTMP协议、RTSP协议、MMS协议等，HLS直播最大的不同在于，直播客户端获取到的，并不是一个完整的数据流。HLS协议在服务器端将直播数据流存储为连续的、很短时长的媒体文件（MPEG-TS格式），而客户端则不断的下载并播放这些小文件，因为服务器端总是会将最新的直播数据生成新的小文件，这样客户端只要不停的按顺序播放从服务器获取到的文件，就实现了直播。由此可见，基本上可以认为，HLS是以点播的技术方式来实现直播。由于数据通过HTTP协议传输，所以完全不用考虑防火墙或者代理的问题，而且分段文件的时长很短，客户端可以很快的选择和切换码率，以适应不同带宽条件下的播放。不过HLS的这种技术特点，决定了它的延迟一般总是会高于普通的流媒体直播协议。

## **二、总结：**

如果大家之前看过我之前的那篇srs的推流和拉流操作：[手把手带你实现srs流媒体推流和拉流操作](https://mp.weixin.qq.com/s/z4bQc7e_A-1xqg-u5EQE-g)，本篇文章就非常简单了，理解一下Nginx的工作原理就行了；最近和网友在交流nginx以及rtmp的延迟问题，这个问题蛮需要实际工作积累才好解决！最后欢迎大家来多多交流！

文章来源：

<https://willarun365.blog.csdn.net/article/details/82877422>